A combined mathematica–fortran program package for analytical calculation of the matrix elements of the microscopic cluster model

Kálmán Varga

Institute of Nuclear Research of the Hungarian Academy of Sciences

(MTA ATOMKI) Debrecen, Hungary

February 5, 2008

Abstract

We present a computer code that analytically evaluates the matrix elements of the microscopic nuclear Hamiltonian and unity operator between Slater determinants of displaced gaussian single particle orbits. Such matrix elements appear in the generator coordinate model and the resonating group model versions of the microscopic multicluster calculations.

PROGRAM SUMMARY 8pt

Title of the program: MCKER

Catalog number:

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue)

Licensing provisions: None

Computer: IBM PC 386DX

Operating system: DOS 3.2

Programming language used: Mathematica v.1.2, Fortran

No. of lines in distributed program,including test data etc.:

Keywords: microscopic nuclear cluster model, generator coordinate method

Nature of physical problem

This computer code contructs matrix elements to be used in microscopic studies of light multicluster system.

Method of solution

We analytically evaluate the determinants and inverses appearing in the matrix elements of Slater determinants of nonorthogonal single particle states, using the Mathematica symbolic manipulation language. A fortran code is used to substitute the space part of the single particle matrix elements.

Restrictions on the complexity of the problem

Varies with available RAM and type of computer. Use of the available RAM depends on the number of clusters and nucleons.

Typical running time

For the example given in the test run input, approximately 300 s.

LONG WRITE-UP

1 Introduction

The microscopic nuclear cluster model provides an unquestionably successful description of light nuclear systems [1], [2]. The main difficulty in performing microscopic cluster-model calculation is the evaluation of the microscopic Hamiltonian between Slater determinants of nonorthogonal single particle-states. This calculation becomes especially tedious for multicluster systems. The need for analytical form of these matrix elements calls for application of symbolic manipulation languages.

In this paper the single particle states of the Slater determinants are displaced gaussian functions, that is, we determine the matrix elements of the "generator coordinate model" (GCM) [1]. To achieve this, we combine the advantage of computer languages the Fortran and the Mathematica [3]: With the help of Mathematica we can derive analytical expressions for the matrix elements, with Fortran we will sort out and specialize the formulae.

Our program is applicable for a general N-cluster system of $0s$ clusters of common oscillator width. The generalization of the program for nonequal width parameters or for higher harmonic oscillator orbits is straightforward. We hope that our program brings the technique of the nuclear cluster model in the reach of nuclear physicist.

2 Formulation of the problem

Let us consider an N-cluster system, where each cluster consists of A_i ($i = 1, ..., N$, $\sum_{i=1}^{N} A_i = A$) nucleons, and the kth nucleon occupies the single particle state $\psi_j(x_k)$ (see Figure 1). The notation x_k stands for the space-, spin- and isospin-coordinate of the kth nucleon: $x_k = (\mathbf{r}_k, \sigma_k, \tau_k)$. We assume that the nucleons of the ith cluster occupy the single particle states of the form

$$\psi_j(x_k) = \varphi_i(\mathbf{r}_k)\chi_{\sigma_j}(k)\chi_{\tau_j}(k), \qquad (j = A_{i-1}, ..., A_i), \tag{1}$$

where $\varphi_i(\mathbf{r}_i)$, $\chi_{\sigma_j}(k)$ and $\chi_{\tau_j}(j)$ are the space, spin and isospin parts of the wavefunction, respectively. (Note the cluster label i on $\varphi_i(\mathbf{r}_i)$.) We restrict ourselves to the case, where the space and the isospin parts of the single-particle function of the nucleons of the A particle system are fixed, but the spins may have different orientation. A given set of spin quantum numbers is labelled by α.

The spin and the isospin functions of different states are orthogonal

$$\langle \chi_{\sigma_k} | \chi_{\sigma_l} \rangle = \delta_{\sigma_k \sigma_l}, \tag{2}$$

$$\langle \chi_{\tau_k} | \chi_{\tau_l} \rangle = \delta_{\tau_k \tau_l}, \tag{3}$$

the space-part of single-particle states overlap, defining an overlap matrix

$$b(i,j) = \langle \varphi_i | \varphi_j \rangle. \tag{4}$$

The wave function of the A-particle system is given by linear combination

$$\Psi = \sum_\alpha a_\alpha \Psi^\alpha \tag{5}$$

of the Slater determinants

$$\Psi^\alpha = \frac{1}{\sqrt{A!}} \det\{\psi_i^\alpha(x_j)\} \tag{6}$$

of the different sets of single particle states.

The matrix elements of Slater determinants can be calculated using well-known rules [4], [5]. In the following we give a concise summary of these formulae to establish the formalism of our program.

The overlap of two Slater determinants can be written as

$$\langle \Psi^\alpha | \Psi^\beta \rangle = \det\{\langle \psi_k^\alpha | \psi_l^\beta \rangle\}. \tag{7}$$

By using eqs. (2), (3) and (4) the overlap of the single-particle states becomes

$$B_{kl}^{\alpha\beta} = \{\langle \psi_k^\alpha | \psi_l^\beta \rangle\} = b(i,j)\delta_{\sigma_k^\alpha \sigma_l^\beta}\delta_{\tau_k \tau_l}, \qquad (k = A_{i-1}, ..., A_i, \quad l = A_{j-1}, ..., A_j), \quad (i,j = 1, ..., N) \tag{8}$$

thus to determine the overlap we must calculate the determinant of the matrix

$$B^{\alpha\beta} = \{B_{ij}^{\alpha\beta}\} \quad (i,j = 1, ..., A). \tag{9}$$

The matrix elements of the one-body operator are given by

$$\sum_{i=1}^{A} \langle \Psi^\alpha | \mathcal{O}_i^{(1)} | \Psi^\beta \rangle = \sum_{i=1}^{A} \sum_{j=1}^{A} \langle \psi_i^\alpha | \mathcal{O}^{(1)} | \psi_j^\beta \rangle \Delta_{ij}^{\alpha\beta}, \tag{10}$$

where $\Delta_{ij}^{\alpha\beta}$ is the cofactor of $B^{\alpha\beta}$ (i.e. the subdeterminant, obtained by crossing out the ith row and the jth column of $B^{\alpha\beta}$ and multiplying it with the phase $(-1)^{i+j}$).

The matrix elements of the two-body operator can be determined from the formula

$$\sum_{i<j}^{A} \langle \Psi^\alpha | \mathcal{O}_{ij}^{(2)} | \Psi^\beta \rangle = \sum_{\substack{i<j \\ k<l}}^{A} \langle \psi_i^\alpha \psi_j^\alpha | \mathcal{O}^{(2)} | \psi_k^\beta \psi_l^\beta \rangle_a (-1)^{i+j+k+l} \mathrm{det} D_{ijkl}^{\alpha\beta}, \tag{11}$$

where

$$|\psi_k \psi_l\rangle_a = |\psi_k \psi_l\rangle - |\psi_l \psi_k\rangle, \tag{12}$$

and where $D_{ijkl}^{\alpha\beta}$ is an $(A-2)\times(A-2)$ matrix built up of elements of $B^{\alpha\beta}$ by leaving out its rows i and k and its columns j and l.

In this paper we limit ourselves to the case of spin-isospin independent one-body operators, and for the two-body operator of the form

$$\mathcal{O}_{ij}^{(2)} = V(i,j)(w + m{P_{ij}}^{r} + b{P_{ij}}^{\sigma} - h{P_{ij}}^{\tau}) \tag{13}$$

where P_{ij} are the space- , spin- , and isospin-exchange operators. These restrictions are not too serious, one can easily generalize the program to allow for other type of operators.

3 Mathematica part

The analytical calculations of the determinants and sums of eqs. (7), (10) and (11) can be easily performed by using the algebraic computer language Mathematica.

By the aid of Mathematica we first evaluate the single particle overlaps (see eqs. (2) and (3)) and then build up the matrices B, Δ^{ij} and D^{ijkl} and carry out the operations in eqs. (7), (10) and (11) . The matrix elements ("kernels") have the form

– overlap :

$$\langle \Psi^\alpha | \Psi^\beta \rangle = \sum_{n=1}^{n_o} c_n^o \prod_{l_n=1}^{m_n} b(i_{l_n}, j_{l_n})^{k_{l_n}}, \tag{14}$$

– one-body :

$$\sum_{i=1}^{A} \langle \Psi^\alpha | \mathcal{O}_i^{(1)} | \Psi^\beta \rangle = \sum_{n=1}^{n_{ob}} c_n^{ob} t(p_n, q_n) \prod_{l_n=1}^{m_n} b(i_{l_n}, j_{l_n})^{k_{l_n}}, \tag{15}$$

where $t(p,q) = \langle \varphi_p | \mathcal{O}^{(1)} | \varphi_q \rangle$,

– two-body :

$$\sum_{i<j}^{A}\langle\Psi^{\alpha}|\mathcal{O}_{ij}^{(2)}|\Psi^{\beta}\rangle=\sum_{\nu=1}^{\nu_{tb}}(wW_{\nu}+mM_{\nu}+bB_{\nu}+hH_{\nu})v(p_{\nu},q_{\nu};r_{\nu},s_{\nu})\sum_{n=1}^{n_{\nu}^{tb}}c_{n}^{tb}\prod_{l_n=1}^{m_n}b(i_{l_n},j_{l_n})^{k_{l_n}}\,, \qquad (16)$$

where $\langle\varphi_{p_\nu}\varphi_{q_\nu}|\mathcal{O}^{(2)}|\varphi_{r_\nu}\varphi_{s_\nu}\rangle = v(p_\nu,q_\nu;r_\nu,s_\nu)(wW_\nu+mM_\nu+bB_\nu+hH_\nu)$. All the coefficients, indices, limits of summations and products are to be determined by Mathematica. A detailed example of these formulae can be found in section 3.3.

3.1 Input data

The input data are to be read from the input file "*nuc.dat*". Detailed input data specification:

1) NumClus: Number of clusters.

2) NumPar: Number of particles in the clusters. (Numpar(i), i=1,...,Numclus)

3) isospin: Charge number of the particles, 1 stands for protons and 0 stands for neutrons, (isospin(i), i=1,...,A).

4) SpinConf: Number of spin configurations.

5) spin: Spin of the particles, 1 stands for spin up, 0 stands for spin down, (spin(i,j), i=1,...,A, j=1,...,SpinConf).

6) a_α: Clebsch-Gordan coefficients of the spin configurations, (α=1,...,SpinConf).

3.2 Output data

To calculate the overlap, the one-body and the two-body kernels, one should use the programs "*ok.m*", "*obk.m*" and "*tbk.m*", respectively. The output is written into the files "*ok.out*", "*obk.out*" and "*tbk.out*", accordingly. As output we get the coefficients and indices of eqs. (14), (15) and (16) ordered according to Table 1. In the output files the results for different spin configurations follow in consecutive order.

3.3 Test run

As a test run, we consider the example of ^{6}He=$\alpha+n+n$. This is a three-cluster system, the two outer neutrons may have two different spin configurations with antiparallel spins (see the test run input). The combination coefficients are $a_1 = -a_2 =< \frac{1}{2}\frac{1}{2}\frac{1}{2} - \frac{1}{2}|00 >= 1/\sqrt{2}$. The package must be loaded while running the Mathematica program. The program should work with any Mathematica implementation on any machine without modifications. The output of "*ok.m*" (ok.out) corresponds to the overlap

$$\begin{aligned}
\langle\Psi|\Psi\rangle &= b(1,1)^2b(1,2)b(1,3)b(2,1)b(3,1) - b(1,1)^3b(1,3)b(2,2)b(3,1) - b(1,1)^3b(1,2)b(2,1)b(3,3) \\
&+ b(1,1)^4b(2,2)b(3,3) - b(1,1)^2b(1,2)b(1,3)b(2,1)b(3,1) + b(1,1)^3b(1,2)b(2,3)b(3,1) \\
&+ b(1,1)^3b(1,3)b(2,1)b(3,2) - b(1,1)^4b(2,3)b(3,2) - b(1,1)^2b(1,2)b(1,3)b(2,1)b(3,1) \\
&+ b(1,1)^3b(1,2)b(2,3)b(3,1) + b(1,1)^3b(1,3)b(2,1)b(3,2) - b(1,1)^4b(2,3)b(3,2) \\
&+ b(1,1)^2b(1,2)b(1,3)b(2,1)b(3,1) - b(1,1)^3b(1,3)b(2,2)b(3,1) - b(1,1)^3b(1,2)b(2,1)b(3,3) \\
&+ b(1,1)^4b(2,2)b(3,3). \qquad (17)
\end{aligned}$$

There are seven different terms, as the 1st, 5th, 9th, 13th, the 2nd, 24th, the 3rd,15th, the 4th, 16th, the 6th, 10th, the 7th, 11th and the 8th, 12th terms of 16 elements of the sum are equal. These will be collected by the Fortran part.

The output of the programs "*obk.m*" and "*tbk.m*" can be understood similarly:

$$\langle\Psi|T|\Psi\rangle = -t(3,3)b(1,1)^3b(1,2)b(2,1) + t(3,1)b(1,1)^2b(1,2)b(1,3)b(3,1) + ..., \qquad (18)$$

and

$$\begin{aligned}
\langle\Psi|\mathcal{O}_{ij}^{(2)}|\Psi\rangle &= (W_1 + M_1 + H_1 + B_1)v(1,1,1,1)(b(1,2)b(1,3)b(2,1)b(3,1) \qquad (19) \\
&- b(1,1)b(1,3)b(2,2)b(3,1) - b(1,1)b(1,2)b(2,1)b(3,3) + b(1,1)^2b(2,2)b(3,3)) + \\
&\quad (W_2 + M_2 + H_2 + B_2)v(1,1,1,1)(b(1,2)b(1,3)b(2,1)b(3,1) \\
&- b(1,1)b(1,3)b(2,2)b(3,1) - b(1,1)b(1,2)b(2,1)b(3,3) + b(1,1)^2b(2,2)b(3,3)) + ...
\end{aligned}$$

As the output of "*tbk.m*" is too long (560 terms), we present only the first 546 lines of "*tbk.out*".

The "-1" at the bottom of the Mathematica's output serves as an end of file indicator for the Fortran part.

4 Fortran part

Due to the spin-isospin symmetry, a great number of terms of the sums (14), (15) and (16) will be exactly equal in the Mathematica's output. By collecting these, the number of terms can be reduced by about an order of magnitude. E.g., in the case of ^{8}He=α+n+n+n+n five cluster system, the potential energy kernel contains about 50000 terms (see eq.(16)),and only 3528 of them are different. As the Mathematica stores the symbols and operations, this collection by Mathematica would require enormously large memory and might be very slow. As in the Mathematica's output the analytical formulae (14), (15) and (16) are represented by integer numbers, this simplification can efficiently be achieved by using the Fortran language.

After collecting the identical terms, the forms of the expressions (14), (15) and (16) remain the same, except for the coefficient c_n^o, c_n^{ob} and c_n^{tb} , where the multiplicity of each different terms appears.

In the Fortran part we can also substitute the concrete form of the spatial part of the overlap, one-body and two-body matrix elements of the single-particle functions. In this paper we limit ourselves to the case of displaced gaussian single-particle functions of equal size parameters:

$$\varphi_i(\mathbf{r}_k) = \left(\frac{2\nu}{\pi}\right)^{3/4} e^{-\nu(\mathbf{r}-\mathbf{s}_i)^2}, \tag{20}$$

where $\mathbf{s}_i$ is the "generator coordinate vector" pointing to the centre-of-mass of the ith cluster (see Figure 1). For the case of nonequal size parameters or for other type of single particle functions this part of the program can easily be generalized. As the generator coordinates in the bra may differ from those in the ket the vectors in the bra and ket position will be distinguished by using a prime ($'$).

The two-body potential is represented by gaussian form factor

$$V(i,j) = e^{-\mu(\mathbf{r}_i-\mathbf{r}_j)^2}, \tag{21}$$

and as an example, the one-body operator is chosen to be the kinetic energy operator T. The single particle matrix elements are [5]

$$b(i,j) = e^{(\mathbf{s}_i-\mathbf{s'}_j)^2}, \tag{22}$$

$$t(p,q) = b(p,q)\frac{\hbar^2}{2m}\nu(3-\nu(\mathbf{s}_p-\mathbf{s'}_q)^2), \tag{23}$$

$$v(p,q;r,t) = \theta^{3/2}b(p,r)b(q,t)e^{-\frac{1-\theta}{4}\nu(\mathbf{s}_p-\mathbf{s}_q+\mathbf{s'}_r-\mathbf{s'}_t)^2}, \tag{24}$$

where $\theta = \nu/(\nu+\mu)$.

By substituting the single particle matrix elements (22), (23) and (24) into eqs. (14), (15) and (16) the matrix elements of the wave function Ψ take the form:

– overlap:

$$\langle\Psi|\Psi\rangle = \sum_{i=1}^{N_o} C_i^o \exp(-\frac{1}{2}\nu s^\dagger A_i^o s), \tag{25}$$

– one-body:

$$\langle\Psi|T|\Psi\rangle = \frac{\hbar^2\nu}{2m}\sum_{i=1}^{N_{ob}} C_i^{ob}(3-\nu s^\dagger B_i^{ob} s)\exp(-\frac{1}{2}\nu s^\dagger A_i^{ob} s), \tag{26}$$

– two-body:

$$\langle\Psi|\mathcal{O}_{ij}^{(2)}|\Psi\rangle = \theta^{3/2}\sum_{i=1}^{N_{tb}}(wC_i^W + mC_i^M + bC_i^B + hC_i^H)\exp(-\frac{1}{2}\nu s^\dagger A_i^{tb} s - \frac{1-\theta}{4}\nu(a_i^{tb} s)^2), \tag{27}$$

where $A_i^o, A_i^{ob}, A_i^{tb}$ and B_i^{ob} are $2N \times 2N$ matrices, a_i^{tb} is an $2N$ dimensional vector, and

$$s^\dagger = (\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_N, \mathbf{s}'_1, \mathbf{s}'_2, ..., \mathbf{s}'_N)\,. \tag{28}$$

After suitable angular momentum projection /citeBrink, these matrix elements are building blocks of the cluster model calculations.

4.1 Input data

The Fortran part needs the input data stored in the file "*nuc.dat*", and the Mathematica's output "*ok.out*", "*obk.out*" and "*tbk.out*". Free format is used.

4.2 Output data

The Fortran programs "*ok.f*", "*obk.f*" and "*tbk.f*" handle the overlap, the kinetic energy and the potential energy kernels, respectively. The output is written in the files "*ok.inp*", "*obk.inp*" and "*tbk.inp*" ordered as shown on Table 2.

4.3 Test run

After collecting the identical terms, the Fortran part substitutes the single particle matrix elements, thus e. g. the overlap looks like:

$$\begin{aligned}\langle\Psi|\Psi\rangle &= 2\exp\{-\frac{\nu}{2}(2(\mathbf{s}_1-\mathbf{s}'_1)^2 + (\mathbf{s}_1-\mathbf{s}'_2)^2(\mathbf{s}_1-\mathbf{s}'_3)^2 + (\mathbf{s}_2-\mathbf{s}'_1)^2 + (\mathbf{s}_3-\mathbf{s}'_1)^2)\} \\ &- \exp\{-\frac{\nu}{2}(3(\mathbf{s}_1-\mathbf{s}'_1)^2 + (\mathbf{s}_1-\mathbf{s}'_3)^2 + (\mathbf{s}_2-\mathbf{s}'_2)^2 + (\mathbf{s}_3-\mathbf{s}'_1)^2)\}\end{aligned}$$

$$\begin{aligned}
-&\quad \exp\{-\frac{\nu}{2}(-3(\mathbf{s}_1-\mathbf{s'}_1)^2+(\mathbf{s}_1-\mathbf{s'}_2)^2+(\mathbf{s}_2-\mathbf{s'}_1)^2+(\mathbf{s}_3-\mathbf{s'}_3)^2)\}\\
+&\quad \exp\{-\frac{\nu}{2}(+4(\mathbf{s}_1-\mathbf{s'}_1)^2+(\mathbf{s}_2-\mathbf{s'}_2)^2+(\mathbf{s}_3-\mathbf{s'}_3)^2)\}\\
-&\quad \exp\{-\frac{\nu}{2}(3(\mathbf{s}_1-\mathbf{s'}_1)^2+(\mathbf{s}_1-\mathbf{s'}_2)^2+(\mathbf{s}_2-\mathbf{s'}_3)^2+(\mathbf{s}_3-\mathbf{s'}_1)^2)\}\\
-&\quad \exp\{-\frac{\nu}{2}(3(\mathbf{s}_1-\mathbf{s'}_1)^2+(\mathbf{s}_1-\mathbf{s'}_3)^2+(\mathbf{s}_2-\mathbf{s'}_1)^2+(\mathbf{s}_3-\mathbf{s'}_2)^2)\}\\
+&\quad \exp\{-\frac{\nu}{2}((\mathbf{s}_1-\mathbf{s'}_1)^2+(\mathbf{s}_2-\mathbf{s'}_3)^2+(\mathbf{s}_3-\mathbf{s'}_2)^2)\}
\end{aligned}$$

From this form the matrices A^o in eq. (25) are determined and written into "*ok.out*" (see the test run output). The output files of "*obk.out*" and "*obk.out*" are too long, and only the first 53 and 57 lines are listed.

Acknowledgements

The first version of this program was written while I was Visiting Scientist at Niigata University. I am very pleased to thank Professor Y. Suzuki for his hospitality. I am grateful to Dr. A. Csótó and Dr. A. T. Kruppa for useful discussions. This work was supported by the OTKA grant (No. F4348).

References

[1] K. Wildermuth and Y. C. Tang, A Unified Theory of the Nucleus (Vieweg, Braunschweig, 1977)

[2] K. Langanke, J.A. Maruhn and S.E. Koonin, Computational Nuclear Physics 2, Nuclear Reactions, Springer-Verlag 1992, p. 70

[3] S. Wolfram, Mathematica – A system for Doing Mathematics by Computer, Addison-Wesley, Redwood City, CA,1988

[4] D. Brink, Proceedings of the International School of Physics,"Enrico Fermi", Course **36** (1965), ed. C. Bloch, (Academic Press, New York and London, 1966), p. 247

[5] H. Horiuchi, Suppl. of Prog. Theor. Phys. **62** (1977) 90.

Table 1

$overlap$	$one - body$	$two - body$
α, β, n_o	α, β, n_{ob}	α, β
m_1	p_1, q_1	W_1, M_1, B_1, H_1
$i_{1_1}, j_{1_1}, k_{1_1}$	m_1	p_1, q_1, r_1, s_1
$\vdots$	$i_{1_1}, j_{1_1}, k_{1_1}$	n_1
$i_{m_1}, j_{m_1}, k_{m_1}$	$\vdots$	m_1
c_1^o	$i_{m_{1_1}}, j_{m_{1_1}}, k_{m_{1_1}}$	$i_{1_1}, j_{1_1}, k_{1_1}$
m_2	c_1^{ob}	$\vdots$
$i_{1_2}, j_{1_2}, k_{1_2}$	p_2, q_2	$i_{m_1}, j_{m_1}, k_{m_1}$
$\vdots$	m_2	c_1^{tb}
	$i_{1_2}, j_{1_2}, k_{1_2}$	m_2
	$\vdots$	$i_{1_1}, j_{1_1}, k_{1_1}$
		$\vdots$
		$i_{m_2}, j_{m_2}, k_{m_2}$
		c_2^{tb}
		$\vdots$
		m_{n_1}
		$i_{1_1}, j_{1_1}, k_{1_1}$
		$\vdots$
		$i_{m_{n_1}}, j_{m_{n_1}}, k_{m_{n_1}}$
		$c_{n_1}^{tb}$
		α, β
		W_2, M_2, B_2, H_2
		p_2, q_2, r_2, s_2
		$n_2 m_1$
		$\vdots$

Table 2

$overlap$			$one-body$			$two-body$		
N_o			N_{ob}			N_{tb}		
C_1^o			C_1^{ob}			$C_1^W, C_1^M, C_1^B, C_1^H$		
$(A_1^o)_{11}$	...	$(A_1^o)_{1N}$	$\left(A_1^{ob}\right)_{11}$	...	$\left(A_1^{ob}\right)_{1N}$	$\left(A_1^{tb}\right)_{11}$	...	$\left(A_1^{tb}\right)_{1N}$
$\vdots$		$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$
$(A_1^o)_{N1}$	...	$(A_1^o)_{NN}$	$\left(A_1^{ob}\right)_{N1}$	...	$\left(A_1^{ob}\right)_{NN}$	$\left(A_1^{tb}\right)_{N1}$	...	$\left(A_1^{tb}\right)_{NN}$
C_2^o			$\left(B_1^{ob}\right)_{11}$	...	$\left(B_1^{ob}\right)_{1N}$	$\left(a_1^{tb}\right)_1$	...	$\left(a_1^{tb}\right)_N$
$\vdots$			$\vdots$		$\vdots$	$C_2^W, C_2^M, C_2^B, C_2^H$		
			$\left(B_1^{ob}\right)_{N1}$	...	$\left(B_1^{ob}\right)_{NN}$	$\vdots$		
			C_2^{ob}					
			$\vdots$					